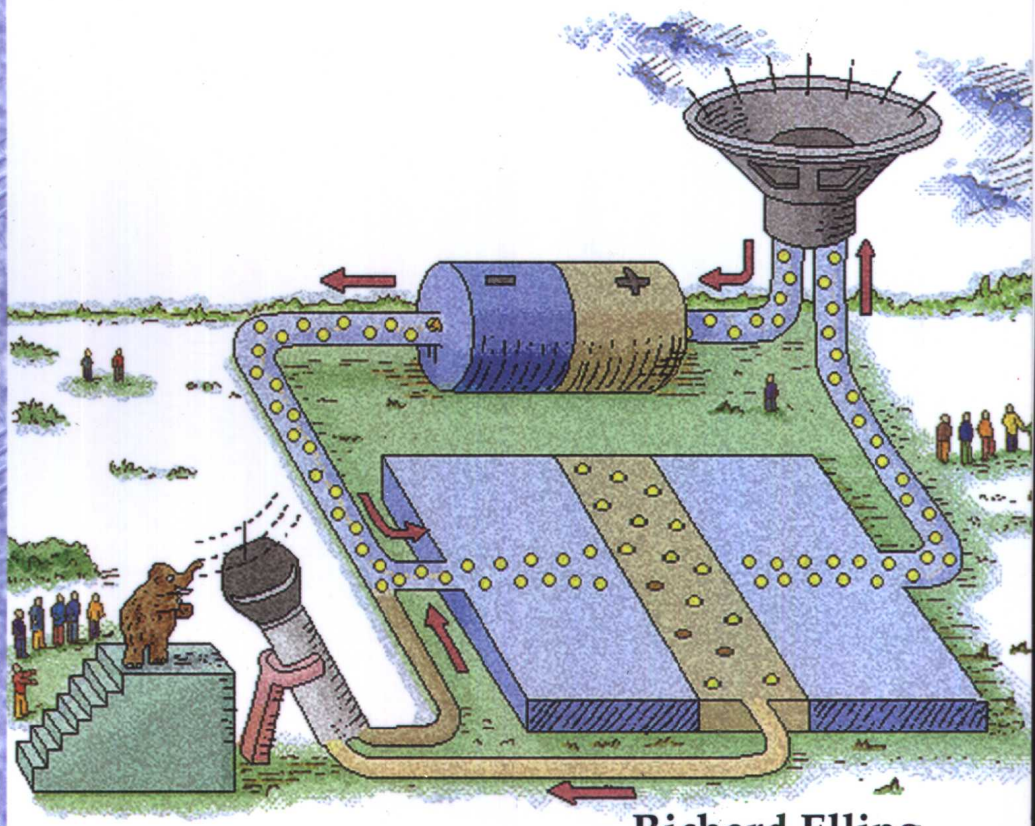
HZ BOOKS

Designing Enterprise Solutions with Sun Cluster 3.0

Sun 公司核心技术丛书

Sun Cluster 3.0

企业解决方案



(美) Richard Elling
Tim Read 著

王建华 王卫峰 译

机械工业出版社
China Machine Press

PH PTR

Sun公司核心技术丛书

Sun Cluster 3.0 企业解决方案

Designing Enterprise Solutions with Sun Cluster 3.0

（美） Richard Elling
Tim Read 著

王建华 王卫峰 译

机械工业出版社
China Machine Press

本书详细介绍了Sun Cluster 3.0软件在企业群集系统中的应用。主要讲解了Sun Cluster 3.0软件的结构、特性以及它在解决数据的同步、数据的高速缓存、数据的备份与群集系统的各种故障等多方面的应用，并且提供了如何使用该软件技术的详细示例。本书可供对群集系统技术具备不同程度的经验和知识的读者阅读。

本书版权登记号：图字：01-2002-3651

图书在版编目（CIP）数据

Sun Cluster 3.0企业解决方案 /（美）伊林（Elling, R.），（美）瑞德（Read, T.）著；王建华，王卫峰译. -北京：机械工业出版社，2002.10

（Sun公司核心技术丛书）

书名原文：Designing Enterprise Solutions with Sun Cluster 3.0

ISBN 7-111-10951-1

Ⅰ. S… Ⅱ. ①伊… ②瑞… ③王… ④王… Ⅲ. 计算机网络-系统管理 - 应用软件，Cluster 3.0 Ⅳ. TP393.07

中国版本图书馆CIP数据核字（2002）第070483号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：李云静

北京市密云县印刷厂印刷 · 新华书店北京发行所发行

2002年10月第 1 版第 1 次印刷

787mm × 1092mm 1/16 · 12.5印张

印数：0 001-4 000册

定价：25.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译 者 序

目前，越来越多的企业都依靠计算机来从事它们的经营活动，并且希望每天24小时都能够得到计算机系统连续不断的服务。虽然计算机系统的可靠性正在不断提高，但是，计算机的运行难免会发生故障。为了提高计算机系统的性能和可用性，许多企业都建立了计算机群集系统，以便在某一台或某几台计算机发生故障时，其他计算机能够取而代之继续运行，从而为企业提供不间断的服务。

为了适应企业建立群集系统的需要，Sun公司推出了群集系统软件Sun Cluster 3.0，为企业的群集系统设计提供了出色的解决方案，用于解决数据的同步、数据的高速缓存、数据的备份以及群集系统各种故障的处理等多方面的问题。Sun Cluster 3.0推出了许多新的全局特性，比如全局硬盘、磁带、CD-ROM、全局文件服务程序和全局网络功能等，并且加强了Sun Cluster 3.0对各种高可用性应用程序的支持。本书对Sun Cluster 3.0软件的特性进行了全面而详细的介绍。

本书共分6章和4个附录。第1章介绍企业建立群集系统时试图解决的各种问题。重点讲述如何进行故障、同步和仲裁的处理等3方面的问题。第2章讲述企业群集系统进行信息处理时使用的基本构件，即文件、数据库、名字服务程序、应用服务程序和Web服务程序等。第3章描述Sun Cluster 3.0软件的体系结构。这是使用Sun公司的软件产品来建立连续可用的服务程序的基础。第4章介绍Sun Cluster 3.0的管理服务器的一个例子。第5章和第6章讲述了两个虚构的实例研究。一个是低成本的文件服务器，另一个是在线数据库服务器，用于指导系统设计师在群集系统设计中对各种方案进行利弊的权衡。

附录A列出了用于新型Sun Cluster 3.0产品的一系列设计检查表。附录B深入介绍新型Sun Cluster 3.0产品的开发过程，并且对Sun Cluster 2.2与Sun Cluster 3.0的特性进行了比较。附录C讲述支持高可用性服务程序的数据中心设计的指导原则。附录D简单介绍系统设计师和系统工程师在设计和分析高可用性系统时可以使用的各种工具。

本书由王建华和王卫峰翻译。译文中的不妥之处敬请读者批评指正。

前　言

本书是在Sun公司的Sun BluePrints项目的支持下出版的。主要针对从事群集系统设计的系统设计师和系统工程师。它描述了群集计算机系统的基本系统工程的概念，并且比较详细地介绍了各种解决方案和它们的利弊。

系统工程是要全面回答系统实际应用中出现的某些问题，而这些问题的答案是建立在科学和技术的基础之上的[Ramo 65]（参见本书后的参考文献）。系统工程师要负责处理好人与工艺流程和技术之间的平衡以及各种复杂的多元化问题。他们要将大量的组件、不必要的模式、不完整的要求、不确定的答案、各种外部条件的可能性、复合系统的测试以及作为技术基础的所有自然科学结合在一起。本书对于特定工程设计方案的介绍只是一带而过，它的重点是讲述群集计算机系统设计中反复使用的各个基本概念。

本书介绍了许多关于如何有效地使用群集系统技术的详细例子，并且提供了关于Sun Cluster 3.0系统（以下简称为Sun Cluster 3.0）的特性和功能方面的信息。

书中贯穿了3个基本概念，即故障、同步和仲裁。在系统设计的所有层次上都要反复谈到这3个概念。

首先，复合系统发生故障的原因往往也很复杂。采用群集系统后，可以防止发生某些这样的故障。当采用和维护群集系统的成本小于因为服务中断而导致的损失时，企业就会使用群集系统。虽然你可以预测群集系统上托管的服务发生故障时的各种情况，但仍需努力设计出非常出色的群集系统，以满足企业的需求。

其次，群集系统使用设备的冗余配置来确保任何单个故障点不会影响对数据的访问。但是，给系统添加设备的冗余配置肯定会带来同步问题，也就是说数据的多个拷贝必须保持同步，否则就会导致混乱。

再次，设备的冗余配置和故障会带来仲裁的问题。假设有两个数据拷贝失去了同步，那么究竟哪个拷贝的数据是正确的呢？同样，当你对数据进行服务操作时，总是希望不会在自己不知道的情况下有人对同样的数据执行了其他的数据服务操作。这些仲裁问题是通过群集系统的基础结构提供的服务程序来解决的。

Sun BluePrints项目

Sun公司的Sun BluePrints项目的目的是为Sun公司的客户提供使用Sun公司的产品在数据中心内建立可靠的、广泛的和安全的信息系统所需要的技术知识。该项目提供了一个框架，用于确定、开发和分配适用于整个Sun公司产品系列的最佳实用信息。从事各种不同领域的技术课题研究的专家都为该程序的制订和实施贡献了力量，并且他们重点解决了信息的范围和作用等问题。

Sun公司的Sun BluePrints项目包括有关的著作、指南和在线论文。通过这些载体，Sun公司

提供了产品的指导、安装和实现等方面的经验，实际应用的环境，以及最近取得突破的技术信息。

若要查看电子月刊《Sun BluePrints OnLine》，请访问网址http://www.sun.com/blueprints。若要得到关于对Sun BluePrints项目的更新信息，请在该站点上进行注册。

本书的读者对象

本书主要供对群集系统技术具备不同程度的经验和知识的读者阅读。本书在介绍Sun Cluster 3.0软件的特性和功能的同时，还提供了许多如何使用该软件技术的详细例子。

阅读本书前你应该具备的条件

你应该熟悉基本的系统体系结构和设计原理，还应该知道Solaris运行环境的管理和维护功能。你也应该了解标准的网络协议和网络拓扑。

本书的内容编排

本书分为6章和4个附录，主要内容如下：

第1章介绍群集系统试图解决的问题。该章的重点放在故障、同步和仲裁这3个方面上。复合系统发生故障的情况往往也很复杂；因此，系统工程师的头脑里首先要考虑到各种故障将会对系统产生什么样的影响。同步是使两样（或更多）东西看起来像一个东西的关键，这对于冗余系统来说是非常重要的。仲裁是个决策过程，这是当一个事件发生或者不发生时系统要做的一项工作。

第2章讲述企业群集计算时使用的基本构件，即文件、数据库、名字服务程序、应用服务程序和Web服务程序等，同时，该章还要介绍群集系统技术为什么能够使这些构件具备很高的可用性和可伸缩性。

第3章描述Sun Cluster 3.0软件的体系结构。这是使用Sun公司的软件产品来建立连续可用的服务程序的基础。Sun Cluster 3.0软件包含许多先进的特性，使得系统设计师能够从服务的角度而不是软件的角度来进行应用程序的设计。

第4章介绍Sun Cluster 3.0的管理服务器举例。该章描述了基础结构的服务程序和一个首先提供这些服务程序的管理服务器。该管理服务器将用于后面各章介绍的群集系统解决方案中。

第5章和第6章讲述两个虚构的实例研究。一个是低成本的文件服务器，另一个是在线数据库服务器。每个实例研究都对企业的业务情况进行了介绍，并且定义了客户的要求。这些解决方案可以用于建立设计中的各个要素的优先级，以便指导系统设计师在系统设计中对各种方案进行利弊的权衡。接着，这两章介绍了系统的设计，讲述了系统设计的方法，并且详细探讨了系统设计师需要进行利弊权衡的某些设计问题。

附录A包含用于新型Sun Cluster 3.0产品的一系列设计检查表。

附录B深入介绍新型Sun Cluster 3.0产品的开发过程，并且对Sun Cluster 2.2与Sun *Cluster 3.0* 的特性进行了比较。

附录C讲述支持高可用性服务程序的数据中心设计的指导原则。

附录D简单介绍系统设计师和系统工程师在设计和分析高可用性系统时可以使用的各种工具。

如何订阅Sun公司的资料

SunDocs项目提供了250多种Sun公司的手册。如果你住在美国、加拿大、欧洲或者日本，就可以通过该项目购买Sun公司的整套手册资料，也可以单独购买某个手册。

如何访问Sun公司的在线资料

通过Web站点docs.sun.com，你可以访问Sun公司的在线技术资料。你可以浏览docs.sun.com站点的存档文件，也可以搜索特定的书名或主题。该站点的URL是http://docs.sun.com。

有关的参考书目

下面这个列表列出了提供非常有用的辅助信息的图书。

标　题	作者和出版社	ISBN号/产品号/URL
Sun Cluster Environment Sun Cluster 2.2	Enrique Vargas, Joseph Bianco, and David Deeths Sun Microsystems Press/Prentice Hall, Inc. (2001)	0-13-041870-6
Backup and Restore Practices for Sun Enterprise Servers	Stan Stringfellow, Miroslav Klivansky, and Michael Barto Sun Microsystems Press/Prentice Hall, Inc. (2000)	0-13-089401-X
System Interface Guide	Sun Microsystems	806-4750-10
Multithreaded Programming Guide	Sun Microsystems	806-5257-10
Sun Cluster 3.0 7/01 Collection	Sun Microsystems	`http://www.sun.docs` 和 AnswerBook2™
Building a JumpStart Infrastructure	Alex Noordergraaf Sun Microsystems	`http://www.sun.com/blueprints`
Cluster Platform 220/1000 Architecture—A Product from the SunTone Platforms Portfolio	Enrique Vargas Sun Microsystems	Sun BluePrints OnLine 文章 `http://www.sun.com/blueprints`

命令中的shell提示符举例

下面列出了C shell、Bourne shell和Korn shell的默认系统提示符和特权用户的命令提示符。

shell	提示符
C shell	*machine _name%*
C shell特权用户	*machine_name#*
Bourne shell和Korn shell	$
Bourne shell和Korn shell特权用户	#

作者简介

Richard Elling是Sun公司负责企业解决方案的主要工程设计师。他曾经担任Sun公司的现场系统工程师5年，并曾荣获1996年Sun公司最佳年度全球现场系统工程师的称号。在他加入Sun公司之前，曾经是Auburn大学工程学院的网络支持经理，还担任过一家新兴的微电子公司的设计工程师，并且在美国国家宇航局（NASA）工作过，为航天飞机任务从事过电子设计和实验集成工作。

Tim Read是Sun公司驻英国联合技术机构的高端系统部的首席顾问。从1985年以来，他一直在英国的计算机行业中工作，并且于1990年加入了Sun公司。他获得了伯明翰大学的天体物理学学士学位。作为大学本科学习内容的一部分，他选修了Sun公司的群集系统方面的课程。目前他从事Sun Cluster软件的教学和写作工作。

目　录

第1章 群集系统和复合系统的设计问题

本章将讲述下列几个问题：

- 企业拥有高可用性群集系统的必要性。
- 影响企业决策的系统故障。
- 设计群集系统时应该考虑的因素。
- 群集系统、同步和仲裁（arbitration）的特定故障模式。

若要了解你究竟为什么要设计群集系统，首先必须懂得企业为什么需要这样一个系统。你对这种系统中发生的复杂的系统故障的了解，将会影响你究竟是否决定使用群集系统，并且将有助于你设计一个系统来处理这样的故障。当你设计群集系统时，还必须考虑数据的同步、仲裁，数据的高速缓存、定时和群集系统的各种故障（比如误建分区（split brain）故障，多实例故障和配置信息过时（amnesia）故障）等方面的问题。

一旦熟悉了使你能够设计整个群集系统所需要的所有基本构件、问题和特性后，就可以分析Sun Cluster 3.0软件提供的各种解决方案，以便了解它将如何满足企业的业务需要，以及备份、还原和恢复等要求的。

本章分为下面几个小节：

- 企业建立群集系统的理由。
- 复合系统中出现的故障。
- 数据同步。
- 仲裁方案。
- 数据高速缓存。
- 超时。
- 群集系统中的故障。
- 小结。

1.1 企业建立群集系统的理由

企业之所以要建立计算机群集系统，目的是要提高系统的性能或可用性。有些产品和技术能够同时提高系统的性能和可用性。但是，推动今天的计算机产业发展的许多群集系统开发工作的重点是提高服务的可用性。

对于数量越来越多的计算机用户来说，计算机因为发生故障而停机，这是个非常严重的问题。计算机的可靠性并没有降低，但是用户现在强调要得到更高程度的可用性。由于越来越多的企业依赖计算机作为它们从事经营活动的基础，因此它们都希望能够每天24小时得到计算机系统连续不断的服务。

系统因为故障而停机，将意味着企业要蒙受经济上的损失，甚至可能是巨大的经济损失。受到这个问题困扰的不仅仅有大型企业客户，工作组，甚至桌面用户，也都需要关键任务的计算功能。今天没有人能够承受系统停机带来的损失。即使是为了对系统进行维护而停机，也必须对停机时间加以精打细算。计算机用户希望系统管理员在执行系统维护任务时，系统仍然能够保持运行状态。

当系统停机带来的损失大于建立群集系统所需要增加的成本时，企业就可以运用群集系统来提高系统的可用性。系统停机带来的损失很难准确地预计。为了帮助预计这种损失，你可以使用风险评估法。

1.1.1 风险评估

“风险评估”是在发生某个事件时确定它会产生什么样的结果的一个过程。对于许多企业来说，它的业务处理过程本身就与它们所依赖的计算机系统一样复杂。这就使得系统设计师的风险评估变得非常复杂。有些经营风险可以用成本来表示，对这种类型的一般风险进行评估是比较容易的。但是，要确定群集系统的成本往往是比较困难的，除非你能够说明实现和支持群集系统的成本能够降低系统停机带来的损失。由于前面这种经营风险可以用实际的费用来计量，而群集系统的风险评估则是在多变量环境中进行的，该环境包含许多概率函数，因此许多人发现它比较容易与某个比例的“系统正常工作时间”（uptime）相关联。

群集系统的作用是设法降低某个故障导致服务中断的可能性，但是它们无法防止服务的中断。不过它们确实能够通过提供一个主机来进行故障的恢复，从而尽可能限制服务中断时间。

为了说明建立群集系统所需成本的必要性而进行的计算，不应该将系统出现故障的可能性假设为零。在这种情况下，可以使用几率理论向最终用户说明这一点。说系统“无故障几率为99 %”，不如说“故障几率为1 %”。不过，为了系统设计的需要，系统设计师必须认真考虑存在1 %故障几率的情况。你在进行设计分析的过程中，始终都必须考虑1 %的故障几率。当你对系统停机的风险进行评估后，就可以进行更加具体的成本估算。

1.1.2 成本估算

企业从事的所有经营活动最终都可以归结为成本。如果资金和时间不受任何限制（时间就是金钱），那么就可以制造和运行完美无缺的系统。可惜，大多数实际制造的系统都要受到资金和时间的制约。

临时性费用开支包括硬件和软件的采购费用、操作员培训以及软件开发等费用。通常来说，这些费用是不会再发生的。用于采购群集系统的临时性硬件费用显然要大于等价的单机系统所需要的费用。软件费用的情况是有所差异的。它包括群集系统软件和必要的各种代理软件的费用。由于购买其他软件需要支付软件许可证协议方面的费用，因此会发生一定的额外成本。有些情况下，软件供应商要求你为群集系统中的每个节点购买一个软件许可证。另一些软件供应商则提供比较灵活的许可证办法，比如每个用户购买一个许可证。

续生（recurring）费用包括经常性的维护合同、消耗材料、能源、网络连接费、环境调节费，支持人员和占地面积等方面所需要支出的费用。

几乎所有的系统设计都必须从经济的角度来考虑它是否合算。简单说来，也就是系统产生的效益是否大于它的成本。如果系统的设计不考虑故障停机的问题，那么经济上是否合算的问题往往是很容易计算的。下面是它的计算公式。

$$P_{lifetime} = R_{lifetime} - C_{downtime} - C_{nonrecurring} - \Sigma C_{recurring}$$

其中：

$P_{lifetime}$是系统的寿命期内产生的赢利。

$R_{lifetime}$是系统在它的寿命期内产生的收入。

$C_{downtime}$是系统停机带来的损失。

$C_{nonrecurring}$是临时性费用。

$C_{recurring}$是续生费用。

在系统设计过程中，这些费用往往是很难准确预计的。不过，对于设计出色的系统来说，这些成本是很容易计算的。

系统停机造成的损失常常可以用系统正常工作时间的赢利来表示。其计算公式是：

$$C_{downtime(t)}=t_{down} \times \frac{P_{uptime}}{t_{up}}$$

其中：

$C_{downtime}$是系统停机时间的损失。

t_{down}是系统运行中断的持续时间。

P_{uptime}是t_{up}期间产生的赢利。

t_{up}是系统的正常工作时间。

对于大多数情况来说，这个计算公式已经可以满足需要了。在这个公式中没有考虑到的一个因素是机会成本。如果Web站点拥有许多竞争对手，并且停止运行了，那么客户就可能转向别的某个Web站点。这种情况是一种很难量化的机会损失。

使用这个公式时存在的一个问题是，P_{uptime}可能成为时间的一个函数。例如，采用一班工作制的工厂只能在上班时间内产生赢利。在工厂不开工的时候，P_{uptime}的值是0，因此$C_{downtime}$也是0。

$$C_{downtime(t)}=t_{down} \times \frac{P_{uptime(t)}}{t_{up}}$$

其中：

当上班时间的t = 0时，那么所有其他时间的$P_{uptime\ (t)} = P_{nominal}$

表示系统停机时的实际损失的另一种方法是，根据系统停机对企业产生的影响来计算其损失。例如，支持呼叫中心的系统可以选择使用用户受到系统停机影响的时间（IUM）而不是受损失的费用金额来表示系统停机造成的损失。如果系统停机影响到1000个用户5分钟的工作，那么IUM的值等于1000个用户乘以5分钟，即5000IUM。这种计算方法的优点是它计算起来比较容易。登录到系统上的用户数量和系统中断运行的时间都是很容易测定的数量。可以协商一个

SLA（service level agreement，称为“服务水平协议”或“服务级别约定”），将服务水平设定为IUM。然后由财务人员将IUM转换成费用金额。

使用IUM的另一个好处是，可以对为用户提供的服务进行计算，而不是测定系统组件的可用性。也可以根据服务的可用性来协商确定SLA，但是它比较难以计算用户转用辅助站点的服务的数据。而IUM则可以很容易转为辅助站点的数据计算，因为它的计算根本不是根据系统组件来进行的。

1.2 复合系统中出现的故障

本节将要介绍复合系统中出现的故障形式和它们所产生的影响。根据本节的介绍，你将会了解到复合系统发生的故障是怎样的复杂。在你设计一个能够进行故障恢复的系统之前，必须懂得系统是怎样发生故障的。

故障是设计高可用性（highly available，HA）系统的系统设计师关注的主要焦点。懂得发生故障的可能性、原因、产生的影响，以及故障的检测和恢复，是建立成功的高可用性系统的关键。具有专业水平的高可用性系统设计专家对于各种复合系统往往经过了多年的研究，具有丰富的开发经验，并且拥有用于设计高可用性系统的各种工具。一般的系统设计师不可能拥有这种工具和经验，但是他们也必须设计这样的系统。值得高兴的是，许多具体的工程设计工作已经由Sun公司这样的供应商做好了，这些供应商能够提供集成式的高可用性系统。

典型的系统设计项目最初只是负责定义“该系统打算用于做什么工作”。设计高可用性群集系统的系统设计师也必须将注意力的重点放在“该系统不打算做什么工作”上。这称为测试不想要的工作方式，当你将分开来能够正确地运行，而组合在一起就不能像预期的那样正确运行的组件集成起来时，就会出现这种情况。后面这种系统设计比前面的系统设计要难得多，而且耗费的时间更多，尤其是在功能测试期间，更是如此。通常的功能测试是试图展示系统能够从事它打算要做的工作。但是同样重要和更加困难的是，试图展示系统不能从事它本来不打算做的工作。

所谓系统的“缺陷”（defect）是指系统不能按照它原定的方式来运行。例如，系统缺陷可能是由于系统的设计、制造不当或者误用而造成的后果，硬件或者软件的组件如果设计得不好，制造得不正确，或者受到了损坏，就会出现缺陷。错误（error）通常是由系统的缺陷造成的，如果不纠正这种错误，就会导致系统发生故障（failure）。

系统的缺陷的例子包括：

- 工厂造成的硬件缺陷——连接器中的引线与导线之间的焊接不正确，从而导致连接器使用时数据的丢失。
- 现场出现的硬件缺陷——被损坏的引线不再能够提供连接，从而导致连接器使用时数据的丢失。
- 现场出现的软件缺陷——因为疏忽而损坏的可执行文件可能导致应用程序运行崩溃。

当组件显示出非预定的行为时，便会出现错误，出现错误的原因可能有下面几个：

- 组件存在缺陷。
- 组件的运行参数超出了它预定的范围。

• 某种其他的原因，例如，虽然是预料中的，但却是随机环境因素所造成的影响。

故障（fault）通常是一种缺陷，不过也可能是个运行不准确的错误，并且应该及时加以处理。在软件故障中，“故障”与软件错误可以是同义词，但是也不一定，比如页面错误，就是这样的一种情况。

高可用性计算机系统并不是永远不发生故障的系统。它们或多或少也会像任何其他系统一样，出现相同的组件故障率。高可用性计算机系统与其他系统之间的差别在于它们对故障的响应方式不一样。你可以将响应故障的基本过程分为5个阶段。

图1-1显示了故障响应的5个阶段：

1) 故障检测。

2) 故障隔离，以便确定故障源和必须修理的组件或现场可更换部件（field-replaceble unit，FRU）。

3) 如果是可以自动恢复的组件，比如能够进行检错和纠错（error checking and correction，ECC）的内存，则进行故障纠正。

4) 故障封锁，使故障不会扩散到其他组件。

5) 系统重新配置，以便你对故障组件进行修理。

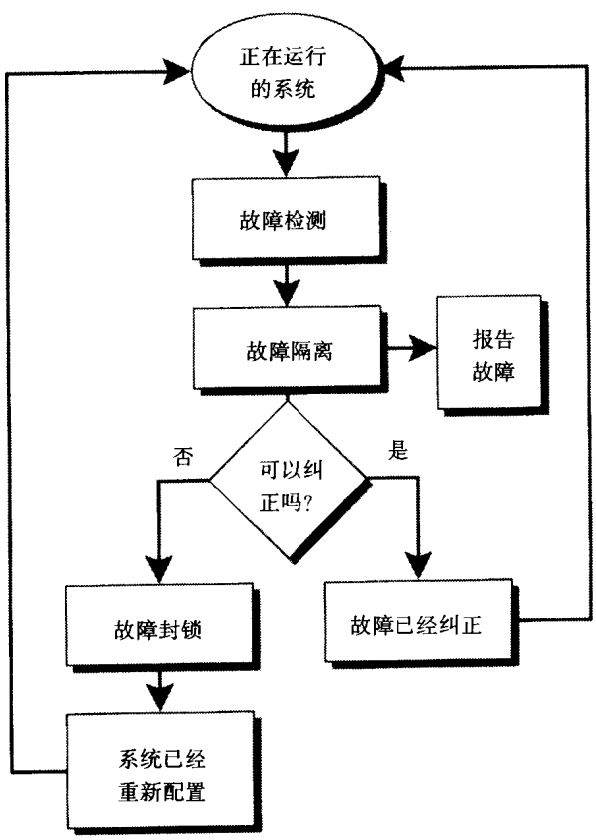


图1-1 高可用性系统故障的响应过程

1.2.1 故障检测

故障检测是高可用性系统的一个重要功能的组成部分。虽然故障检测看起来很简单很容易，但是它也许是群集系统中最复杂的一部分功能。群集系统中故障检测的问题是个开放的问题，对于这个问题，你无法知道所有的解决方案。Sun公司的群集系统解决这个问题的策略是依靠群集系统各个组件之间的业界标准界面。这些界面具有内置的故障发现和错误报告特性。不过，它不可能知道所有组件的所有故障模式及其相互之间的影响。

虽然这听起来是个非常严重的问题，但是当我们懂得了群集系统的运行环境后，情况就不会那样糟糕了。例如，以非屏蔽双绞线（unshielded twisted pair，UTP）10BASE-T以太网接口为例。有两类故障会对该接口产生影响，即物理故障和逻辑故障。这些故障可以根据TCP/IP协议组的4个层作进一步的分类或者确定，不过为了介绍方便起见，我们仅仅进行物理故障和逻辑故障这样的分类就够了。这些故障常常可以由网络接口卡（network interface card，NIC）来进行检测。但是，并不是所有的物理故障都可以由单个网卡来加以检测的，也不是所有物理故障都可以通过简单地去掉一根电缆就能够进行故障模拟的。

懂得系统软件如何在出现故障时对它们进行检测和处理，这对于系统设计师来说是非常重要的。如果网络出现了某种故障，那么软件应该能够将故障确定为究竟是哪个组件造成的故障。

例如，表1-1列出了10BASE-T以太网常见的一些故障形式。从该表中你可以看出，潜在的故障形式很多。其中的有些故障形式是不容易发现的。

表1-1 10BASE-T以太网的常见故障形式

故障形式	故障类型	检测故障的手段	可检测性
电缆没有插好	物理故障	NIC	能，但前提是Software Query Enable（软件查询功能，SQE）必须激活
电缆短路	物理故障	NIC	能
电缆连接的极性相反	物理故障	NIC	能
电缆太长	物理故障	NIC（只能用于某些场合）	很难检测，因为这种故障的形式很多，有的是无链路（SQE功能没有激活），有的是误码率（bit error rate，BER）比较高，必须通过逻辑测试方法才能检测到
电缆的接收对线路的连接存在故障	物理故障	NIC	能，但前提是SQE功能必须激活
电缆的发送对线路的连接存在故障	物理故障	远程设备	能，但前提是SQE功能必须激活
电磁干扰（EMI）	物理故障	NIC（只能用于某些场合）	很难检测，因为这种故障是间隙性的，只有当误码率发生变化时才能检测到
介质访问控制（MAC）地址相重	逻辑故障	Solaris操作系统环境	能
IP地址相重	逻辑故障	Solaris操作系统环境	能
IP网络地址不正确	逻辑故障		一般情况下不能自动检测到
远程主机没有响应	逻辑故障	Sun Cluster软件	Sun Cluster软件使用一系列顺序测试方法，设法建立与远程主机之间的连接